Android VI

Databases
Relational Databases

- Organized into tables that may or may not be related to each other
- Each row in a table is a record that roughly corresponds to a model object
CRUD operations

- Create
- Read
- Update
- Delete
  - The last three may need selection criteria
Steps to create a database in Android

- Create a contract class
- Implement SQLiteOpenHelper
- Implement crud methods
- Use crud methods
- Close databases and cursors
**The Contract**

- An optional class that stores constants we will use in our database operations
- Inner classes represent tables

In our SQLite table, columns represent data represented in UI (non-derived)
SQLiteOpenHelper

- Version number constant (passed into constructor super)
- Database name constant (passed into constructor super)
- Constructor that takes a context, database name, version number
- OnCreate: where the SQL is put to create the tables
  - Use the constants in the contract class
  - Add NOT NULL constraints (to prevent errors due to foreign keys)
- OnUpgrade: called when version number changed and passed into constructor --- manage schema changes here
  - Just “drop tables if exists” for now.
**Crud method read: Query**

`Cursor query (String table, String[] columns, String selection, String[] selectionArgs, StringgroupBy, String having, String orderBy)`

Query the given table, returning a `Cursor` over the result set.

SQLiteDatabase's query method takes in typical arguments in a query and returns a cursor that can be moved to different rows in the result.
CRUD: QUERIES --- JOINS?

```java
Cursor rawQuery (String sql,
    String[] selectionArgs)

Runs the provided SQL and returns a Cursor over the result set.

private final String MY_QUERY =
"SELECT * FROM table_a a INNER JOIN table_b b ON "
+ " a.id=b.other_id WHERE b.property_id=?";

db.rawQuery(MY_QUERY, new String[]{String.valueOf(propertyId)});
```
CRUD METHODS: CREATE, UPDATE

execSQL

void execSQL (String sql)

Execute a single SQL statement that is NOT a SELECT or any other SQL statement that returns data.

- Use execSQL to create tables

update

int update (String table,
            ContentValues values,  
            String whereClause, 
            String[] whereArgs)

Convenience method for updating rows in the database.

- ContentValues are data structures with key/value pairs
- These are used in inserts as well
CRUD METHODS: INSERT, DELETE

**insert**

```java
long insert (String table,
            String nullColumnHack,
            ContentValues values)
```

Convenience method for inserting a row into the database.

**delete**

```java
int delete (String table,
            String whereClause,
            String[] whereArgs)
```

Convenience method for deleting rows in the database.
Working with the recycler view

● We won’t pass a data structure (like an ArrayList of model objects) into the constructor of the adapter
● Instead, we run a query, get a cursor, then pass that into the constructor of the adapter
● Use the cursor in your holder’s bind method to get data for each row
Checking what’s in your emulator’s db

- Make sure adb is in your machine’s path (do it now!)
- Open your command prompt or terminal
- type adb shell
- navigate to your database’s folder (data/data/yoapp/databases)
- type sqlite3
- open database (.open dbfilename)
- run sql commands